

Key Management and ANSI X9.44

**Burt Kaliski, RSA Laboratories
February 10, 2000
NIST Workshop on Key Management
Using Public-Key Cryptography**

- ***Key Establishment Using Factoring-Based Public Key Cryptography for the Financial Services Industry (draft)***
- **Editor: Bob Silverman**
- **Scope: Management of symmetric keys with public-key techniques based on the integer factorization problem**
- **Latest draft: January 2000**

- Key pair generation
- Cryptographic primitives
- Encryption scheme
- Auxiliary functions

- RSA key pairs
 - public key: (n, e)
 - private key: (n, d)
 - where $n = p q$, e odd, $d = e^{-1} \bmod \text{lcm}((p-1), (q-1))$
 - key size: 1024, 1280, 1536, ... bits
- Rabin-Williams (RW) key pairs
 - as above, except:
 - $p \equiv 3 \bmod 8$, $d \equiv 7 \bmod 8$
 - e even, $d = e^{-1} \bmod (\frac{1}{2} \text{lcm}((p-1), (q-1)))$
- Prime generation via ANSI X9.80

- **IFEP1: RSA Encryption**
 - $c = m^e \bmod n$
 - m = message representative, c = ciphertext
- **IFDP1: RSA Decryption**
 - $m = c^d \bmod n$
- **IFEP2: RW Encryption**
- **IFDP2: RW Decryption**

- **ES-OAEP: Encryption with Optimal Asymmetric Encryption Padding**
 - based on Bellare-Rogaway (1994); compatible with IEEE P1363, PKCS #1 v2.0
 - provably secure in random oracle model
- **Encryption operation:**
 - $c = \text{IFEP}(m)$ where $m = \text{OAEP-ENCODE}(M, P)$
 - M = message
 - P = encoding parameters (opt.)
- **Decryption operation:**
 - $M = \text{OAEP-DECODE}(m, P)$ where $m = \text{IFDP}(c)$

- Hash function: SHA-1
- Mask generation function: MGF1
- (Key construction functions currently in annex)

- Security requirements
- Annexes:
 - random number generation [→ ANSI X9.82]
 - key pair generation [→ ANSI X9.80]
 - implementation considerations
 - examples
 - ASN.1 syntax
 - example key management protocols
 - mathematical background [→ ANSI X9.31, X9.80, etc.]

- **Current ANSI X9.44 specifies an encryption scheme, but no key management protocols**
 - (except informative examples in annex)
- **But scope includes symmetric key management**
- **How much further to go?**
- ***Many* possible key management protocols based on ANSI X9.44 encryption scheme**
 - some are still research topics

- **Following IEEE P1363 classification:**
- **A *scheme* is a set of related cryptographic operations**
 - e.g., encryption scheme, signature scheme, key agreement scheme, identification scheme
- **A *protocol* is a sequence of operations to be applied by two or more parties**
 - e.g., entity authentication protocol, key establishment protocol (or combination)
 - may involve operations from more than one scheme

Scheme Standards

- ANSI X9.30:1, X9.31, X9.62
- ANSI X9.42, X9.44 (?)
- FIPS 186-2
- IEEE P1363
- ISO/IEC 9796-1, -2, -3
- ISO/IEC 14888-3

Protocol Standards

- ANSI X9.63
- ANSI X9.70
- FIPS 196
- Key management FIPS
- ISO/IEC 9798-3
- ISO/IEC 11770-3
- ... also, IKE [IPsec],
SSL / TLS, S/MIME /
CMS key management

- How many parties?
- How many key pairs?
- When to generate key pairs?
- How to distribute public keys?
- What is message M ?
- What are parameters P ?
- What else is needed?
 - signature scheme?

- **What are the application requirements?**

- one-pass?
- responder key pair only?
- computational load?

- **What are the security goals?**

- implicit key authentication?
- key confirmation?
- key control?
- replay protection?
- forward secrecy?
- entity authentication?
- etc.

- **Applications using key management:**

- S/MIME / CMS (mail / message security)
- SSL / TLS (session security)

- **Key management standards:**

- ISO/IEC 11770-3
- ANSI X9.70

- Alice needs to transport a content encryption key K to Bob in one pass
- Protocol:
 - (subset of ISO/IEC 11770-3 KT1)
 - A: $c = E_B(K)$
 - A \rightarrow B: c
 - B: $K = D_B(c)$
- Current encryption scheme is PKCS #1 v1.5 or ANSI X9.42 variant; OAEP indicated for future

Implicit key authentication:	B
Key confirmation:	none
Key control:	A
Replay protection:	none
Entity authentication:	none
Forward secrecy:	A

- Alice needs to establish a session key K with Bob but only Bob may have a public key

- Protocol:

A: $c = E_B(\pi)$

A \rightarrow B: c, R_A

B: $\pi = D_B(c); K, K' = \text{KDF}(\pi, R_A, R_B)$

B \rightarrow A: $R_B, \text{MAC}_{K'}(2, B, A, R_B, R_A)$

A \rightarrow B: $\text{MAC}_{K'}(3, A, B, R_A, R_B)$

- where π, R_A, R_B are random

Implicit key authentication: B

Key confirmation: both

Key control: both

Replay protection: both

Entity authentication: B

Forward secrecy: A

- ***Information technology -Security techniques - Key management - Part 3: Mechanisms using asymmetric techniques (draft)***
- **Editor: Xuejia Lai**
- **Scope: Key management mechanisms based on asymmetric cryptographic techniques, including:**
 - symmetric key agreement
 - symmetric key transport
 - public key distribution

- **Seven key agreement mechanisms**
- **Six key transport mechanisms**
- **Abstraction of underlying schemes**
 - key agreement, encryption, and/or signature schemes
 - possibly from different families
 - may include ANSI X9.44 encryption scheme
- **Many variations, different attributes:**
 - one-pass, two-pass, three-pass
 - implicit key authentication, key confirmation, forward secrecy, ...

- ***Management of Symmetric Keys Using Public Key Algorithms (draft)***
- **Editor: Rich Ankney**
- **Scope: Protocol elements for establishing symmetric keys using ANSI-approved public key algorithms, for interactive (session-oriented) key management**
 - store-and-forward key management addressed in ANSI X9.73, Cryptographic Message Syntax

- **Seven key agreement mechanisms**
- **Five key transport mechanisms**
- **One “hybrid” mechanism**
- **Abstraction of underlying schemes**
- **Similar variety to ISO/IEC 11770-3**

- **ANSI X9.44 provides a cryptographic tool for key management**
 - encryption scheme, not yet management protocol
- **Example key management standards provide a useful model**
 - abstraction of underlying schemes
 - multiple protocols from multiple families
- **Industry practice important to consider**
- **Bigger questions: application requirements, security goals**

- | | |
|---------------------|---|
| • ANSI | American National Standards Institute |
| • ANSI X9.31 | Digital Signatures using Reversible Public Key Cryptography (rDSA) |
| • ANSI X9.42 | Agreement of Symmetric Keys using Discrete Logarithm Cryptography |
| • ANSI X9.62 | The Elliptic Curve Digital Signature Algorithm (ECDSA) |
| • ANSI X9.63 | Key Agreement and Key Transport using Elliptic Curve Cryptography |

- **ANSI X9.70** **Management of Symmetric Keys using Public Key Algorithms**
- **ANSI X9.80** **Prime Number Generation, Primality Testing and Primality Certificates**
- **ANSI X9.82** **Random Number Generation**
- **ASN.1** **Abstract Syntax Notation 1**
- **CMS** **Cryptographic Message Syntax**
- **ES-OAEP** **Encryption Scheme using OAEP**
- **FIPS** **Federal Information Processing Standard**

- **FIPS 186-2** **Digital Signature Standard (DSS)**
- **FIPS 196** **Entity Authentication using Public Key Cryptography**
- **IEEE** **Institute of Electrical and Electronics Engineers**
- **IEEE P1363** **Standard Specifications for Public Key Cryptography**
- **IFDP** **Integer Factorization Decryption Primitive**
- **IFEP** **Integer Factorization Encryption Primitive**

- **IKE** **Internet Key Exchange**
- **Ipsec** **Internet Protocol Security**
- **ISO/IEC** **International Standards
Organization/International
Electrotechnical Commission**
- **ISO/IEC 9796-1, -2,-3** **Digital Signature Schemes -
Giving Message Recovery**
- **ISO/IEC 9798-3** **Entity Authentication using a
Public Key Algorithm**
- **ISO/IEC 11770-3** **Key Management : Mechanisms
using Asymmetric Techniques**

- **ISO/IEC 14888-3** **Digital Signatures with Appendix**
- **OAEP** **Optimal Asymmetric
EncryptionPadding**
- **OAEP-DECODE** **OAEP decoding operation**
- **OAEP-ENCODE** **OAEP encoding operation**
- **SHA-1** **Secure Hash Algorithm 1**
- **S/MIME** **Secure Multipurpose Internet
Mail Extensions**
- **SSL** **Secure Sockets Layer**
- **TLS** **Transport Layer Security**